

REC'D 04 APR 2005



WIPO Europäisches PTO
Patentamt

European
Patent Office

Office européen
des brevets

10/591529

73/05/50380

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

04100368.2



**PRIORITY
DOCUMENT**
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk



Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Anmeldung Nr:

Application no.: 04100368.2

Demande no:

Anmelde tag:

Date of filing: 02.02.04

Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Koninklijke Philips Electronics N.V.
Groenewoudseweg 1
5621 BA Eindhoven
PAYS-BAS

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

RCS-1 Dynamic UPnP playlists for continuous playback

In Anspruch genommene Priorität(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)

Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

H03J9/00

Am Anmelde tag benannte Vertragstaaten/Contracting states designated at date of
filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL
PT RO SE SI SK TR LI

RCS-1 Dynamic UPnP playlists for continuous playback

Abstract: The UPnP playback protocol is currently limited to playback of a single item or a playlist of items. Most people will want the device to continue playing the next track after it finished. In normal operation the UPnP controller must handle this itself. In case the controller goes to sleep, playback stops after the current song. This invention describes an idea to overcome this problem.

Background of the invention

The UPnP protocol and architecture is an network-centric way of interacting with content. It distinguishes between UPnP Server (or Content Directory Service), a device (CE device, PC, ..) that contains all kinds of content like music, movies, pictures, games, The UPnP renderer specializes in playing back the content. Finally, the UPnP Control Point provides the UI of the user and contains the intelligence to playback all items on the UPnP network. It allows to user to find and locate content from the different UPnP CDSs and play them back on the available UPnP renderers. It also will show the status of the rendering (i.e. music title, artist, playing time, next song, slideshow, ...).

These logical functions can be distributed on one or multiple devices. Traditionally there will be one device that acts as the UPnP Control Point. Since remote controls are almost always portable, also many UPnP Control Points will be portable.

Problems or disadvantages overcome by the invention

Portable embedded devices have some disadvantages. The characteristics of UPnP put heavy demands on the UPnP controller. It should stay on the network to track UPnP events, have big processing power and have enough memory to deal with a large set of content.

Since many UPnP Control Points are battery operated, this can turn into a problem. In addition to the UPnP architecture implies that a control point is the only active part that handles playback of all content (i.e. start playback of the next track when the current one ends) the UPnP Control Point is expected to stay on. Remaining always on on a wireless network is very power consuming.

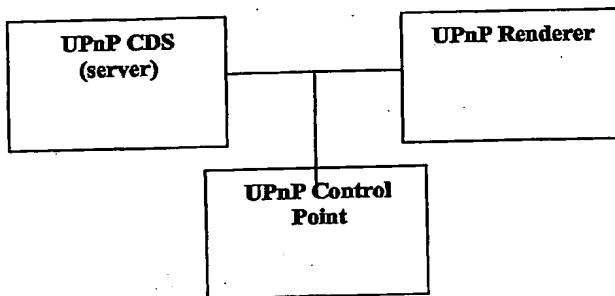
Still, if the Control Points stops running, the continues playback is stopped. This is certainly not what a user wants.

A context of this invention is a feature like 'context switch' to overcome this issue. By this, the embedded sends the context of content-playback to the UPnP renderer at certain times. The render will take over the UPnP control function, so the embedded controller can go to sleep. This idea is a generalization of the 'context switch' idea.

Some feature(s) of the invention

UPnP supports the notion of playlists. A playlist is a list of items that can be played back on an UPnP renderer that supports playlists. Normally customers manually generate playlists. This invention shows a mechanism that automatically can generate a playlist based on the context a user presses on a play button.

Detailed description of how to build and use the invention
The logical distribution of the UPnP architecture.



Below two related flows that use the concept of automatically generated playlists are shown.

A. Automatic playlists handled by the UPnP Control point

1. On the Control Point, the user browses through the CDS tree. This is a tree with all content items. The leaves in the tree are individual items like (music, movie, picture, game, ..). The branches in the tree are logical containers like Album, Genre, Artist, etc.
2. The user can hit a) play on any item in the list of items or b) hit play on a container.
3. In a) the control point constructs a playlist with the item and all items after it and creates a playlist with all those items. Non-playable items (depending on the types the renderer supports are omitted). This playlist keeps order. In b) the control point constructs a playlist with all 'leave' items at the end of the selected branch.
4. A standard UPnP method is used to send the playlist to the renderer. (For this, the UPnP control point might need to implement some functionality of a generic UPnP CDS).
5. The renderer starts playing back the playlist and the user can see details of the playback (time, current artist, song, ...) on the screen of the control point (and renderer).
6. After user-inactivity (or other trigger) the control point can go to any energy-saving mode (or fully off). Still, the playback will continue.

Requisites for this are: a UPnP renderer that supports playlist, a common playlist format, control point that supports this algorithm.

B. Automatic playlists handled by the UPnP CDS

1. On the Control Point, the user browses through the CDS tree. This is a tree with all content items. The leaves in the tree are individual items like (music, movie, picture, game, ..). The branches in the tree are logical containers like Album, Genre, Artist, etc.
2. The user can hit a) play on any item in the list of items or b) hit play on a container.

3. The control point sends this action to the UPnP CDS.
4. In a) the CDS constructs a playlist with the item and all items after it and creates a playlist with all those items. Non-playable items (depended on the types the renderer supports are omitted). This playlist keeps order
In b) the CDS constructs a playlist with all 'leave' items at the end of the selected branch.
5. The CDS returns an identifier of the newly constructed playlist to the Control point.
6. The Control point sends a standard play event to the renderer. The playback url contains the identifier of the new playlist.
7. The renderer starts playing back the playlist and the user can see details of the playback (time, current artist, song, ...) on the screen of the control point (and renderer).
8. After user-inactivity (or other trigger) the control point can go to any energy-saving mode (or fully off). Still, the playback will continue.

Requisites for this are: a UPnP renderer that supports playlist, a UPnP CDS that implements parts of this algorithm, a common playlist format, control point that supports it's part of this algorithm.

General algorithm

Commonly this invention includes an algorithm where a device in the UPnP network can automatically create a playlist based on the context a user presses a play button. There are means that the UPnP renderer can start playing the newly generated playlist.

Applications of the invention

This invention can be used to make UPnP Controller that works behaves how a users will expect it to. Even when it needs to go in a power-saving mode, the playback is continues and as intended by the user.

Background to this invention:

- NL 030821 filed in Europe June 30, 2003, filing no. 03101949.0

Title: EMBEDDING A UPnP AV MEDIASERVER OBJECT ID IN A URI

Inventors: Tim Froidcoeur; Marc Masschelein; Stefaan Motte Daniel Meirsman

Abstract: On a UPnP AV network, different users are identified based on respective IP addresses in the SOAP requests for interaction with AV content stored on the network's MediaServers. Under control of the identity thus determined, the relevant MediaServer generates personalized views of the available content, possibly re-organizing content items in the inventory overview or blocking items from being viewed by specific users on the network.

RCS-2 Web-based method for obtaining station-channel number associations

Abstract: The described method provides a convenient web-based way of associating TV line-up stations to channel numbers, which then allow more convenient tuning of tuning devices by means of a touch-screen remote control unit.

Background of the invention

TV line-ups

Typically cable TV or satellite subscribers receive a series of TV stations (e.g. BBC1, BBC2, NED1, NED2, ARD, ZDF, TF1, FR2, ...) via their cable or satellite set top box. The cable or satellite operator assigns to each station a part of the bandwidth on the transmission medium (e.g. part of the spectrum on a coaxial cable). These bandwidth portions are called channels, identified by a channel number. In order to watch the programs broadcasted by a particular station, a TV's tuner must tune to the correct frequency (range), i.e. bandwidth portion.

In the US, channel numbers for the stations being broadcasted are fixed, that is the user cannot change the channel number that goes with a certain station. In Europe the situation is different: European televisions allow one to associate channel numbers (from 0 to a maximum value, e.g. 99) to stations. That is the station's signal still occupies a given fixed portion of the spectrum, but the user can choose the channel number that identifies that portion.

Touch-screen remote control

The notion exists of touch-screen remote control devices (e.g. Pronto) that have screens holding control buttons representing the digits used for channel selection on a TV, i.e. selecting the appropriate channel number. Pressing these buttons will lead to IR being emitted to the receiving device (the TV in this case) in order to make the built-in tuner tune to the channel of choice.

Problem

When using a touch-screen remote control, users control the TV by means of the TV control pages. That is they press a key sequence of digit keys that will make the TV tune to the channel of their choice. Depending on the channel selection mechanism implemented by the vendor, this may involve 1, 2, or 3 digits, an 'enter' key, a "-/-" key, a "100+" key. This process may be lengthy and require users to memorize station-channel number associations.

Solution

Solution 1

The above process of pressing digit key sequences for selecting a channel number can be made more user-friendly by providing the user with a list of station call signs (and/or logos) that he/she can select from on the remote's touch screen. In order for this to work for a given user, the station-channel number associations that apply to the user's situation need to be known. To that end an internet-service shall be set up that allows the user to go through a personalization process, during which the user is to indicate:

- in the event users can choose channel numbers (e.g. in Europe):

- o what stations he/she receives
- o indicate what channel number apply to the various stations
- in the event of fixed station-channel number associations (e.g. US):
 - o select his/her cable TV or satellite operator and the line-up that he/she receives: this determines the station-channel number associations as these are fixed by the operator for every line-up.

Once the user has provided this information the station-channel number associations (and possibly logos, for display on the remote's touch screen) are downloaded to the user's remote control unit and stored in an on-board database. When a user selects a station logo or call sign, the corresponding channel number is looked up in the on-board database and a macro is launched that will sequentially send commands (eg via IR) to the tuning device corresponding to the sequence of digits and other keys that will make the tuning device tune to the appropriate channel.

Additional simplification

A further simplification can be made for the European situation, by collecting zip codes and cable/satellite operator names from the user during the personalization process, and associating the user's station selection to the operator and area provided. This way after some time it is sufficient to ask zip code and let the user choose from a list of operators. The station list will then be given without the user having to go through a lengthy process of explicitly selecting all stations. As it is the user's initially providing this information themselves this is a very cost-efficient way of collecting such data.

Benefit of the invention

Solution 1: The described method provides a convenient web-based way of associating TV line-up stations to channel numbers, which then allow more convenient tuning of tuning devices by means of a touch-screen remote control unit.

Additional simplification: The described method provides a data collection mechanism for zip codes and cable/satellite operator names, in which users provide the data. This allows the party setting up the web-based service to collect data very cost-efficiently. For reference: In Europe, the sole pan-European data provider Luxemburg-based Infomedia, a Gemstar-owned company, provides Electronic Program Guide data. They have lists of stations available per country. However they do not have regional information (no information on operators for a given zip code, and local TV line up offering of that operator). Hence, the above method could prove very useful.

Possible Embodiments

Any remote control application (hardware product, or computer software (PC, PDA, ...))

RCS-3 Method For Fast Convergence in Search For Infrared Protocol

Abstract: Improved mechanism to find a control protocol in a multitude of possible solutions by determining the most discriminating factors between the remaining candidate protocols.

Prior Art

Partly this application is an extension to Philips patents:

- WO9800933A1 - Remote controller
- US5819294A1 - Automatic configuration mechanism for universal remote

Both describe a method where a user is asked to learn IR control codes, and the system tries to find a match within a set of possible control protocols residing in a database (database can be in RC, CE device, PC, web server)

Problem

For single brand or manufacturer, multiple control protocols exist.

Asking a user which brand TV he has. Usually is not sufficient to find a single solution. Typical big brand names have around 10 entries in the databases.

Result: user sees "Philips 1", "Philips 2", "Philips 3" etc - be it on screen of a remote control or in a booklet or a list on web service application. User has no clue as to how he can know which one is the one right solution for him. He can only find by trial and error.

Some remotes offer "scan" and "try" mechanisms to assist the user, but this only a form of minor automation in an essential trial and error mechanism. The full list is being scanned, solutions get only eliminated one by one, no adaptive search is used. Other databases try to inventorize type numbers of CE equipment. Based on the type number of a users TV (or of the RC that came with it), immediately the right solution is suggested to a user. Big drawback with this solution is that an average consumer does not know the complex type number of his TV (eg on back plate, on user manual..)

Solution

The main principle of the solution is to look at the set of possible matching IR protocols and find the discriminating factors to allow fast convergence. E.g if 10 solutions exist for Philips TV, the method will allow to converge faster than an exhaustive trial and error.

By using non-adaptive trial error, worst case a user will always have to try $N-1$ codes in case of N possibilities.

By using the most discriminating factors, it was demonstrated that convergence can be guaranteed after only 3 or 4 steps, on a representative database of IR codes.

Assumption: user mentions which device type (eg TV) and which Brand the product is to already limit the search space. Brand is not necessarily needed, but does reduce search space dramatically.

Finding most distinctive elements

Most distinctive element = codes that are most likely to differ in implementation between protocols; OR that are present in some and not present in others

This can happen in two ways:

1. static: by analysing a database up front, the on average most distinctive command codes can be determined.
2. dynamic. While a user is searching for certain command protocol, the system analysis the difference between the remaining candidates and extracts most discriminating features (only possible on high processing power devices)

Application on learn & match principle

When a user has his original remote, learn & match (ref to prior art) can be used to speed up convergence.

For a start the statically determined most discriminating functions should be used to immediately eliminate as much as possible codesets from the database.

Once the first cut has been made a dynamic solution could be used.

Note: UI for learn and match should ask a user to learn a certain function from his original remote; but should also allow a user to indicate that a certain function is simply not existing on his RC (in case discriminating code is one that is not implemented in some of the protocols)

Application on database selection

In some cases the user has no original RC, or the original RC does not work correctly anymore.

In that case he needs to search through the database.

The most discriminating factors can be applied to make this search converge faster as well.

Initially user is asked to mention device type and brand to restrict search space.

With the remaining solutions, the system will suggest the most discriminating functions, and transmit them (=try method) to the equipment, asking the user whether the code worked yes/no.

Based on the user's answers, again code sets or protocols can be eliminated quickly, reaching fast convergence in the user's quest for the right control codes.

Benefit of the invention

Solves the problem described earlier: faster convergence in search of Control codes in a database.

Possible Embodiments

Universal remote controls (handheld, probably LCD touchscreen, networking interface) combined with UPnP control point.

Database can reside:

- in the remote control (RC with LCD)
- in a PC (remote with connection to PC); application specific software
- on a webserver (RC connects through network or via PC)

Examples: Pronto, iPronto, WiFi enabled PDA with remote control software,...

RCS-4 IR Connectivity database

Abstract: Universal remote controls are still very complex to set-up. This invention describes a way to ease the setup process of remote controls on a wide variety of aspects. The result is that the remote control becomes more intelligent yet the setup easier. This is achieved by using a 'connectivity database', that contains a whole range of meta-data related to remote controlling CE equipment. This database drives the setup and usage of the remote control.

Background of the invention

The range of different CE devices is still growing fast. Next to traditional TVs, Amplifiers, VCRs, DVDs, new kinds of devices like Web TV, Home cinemas, PC media extenders (play PC music on CE equipment), Internet content players,... are emerging. Also the traditional IR control is getting replaced by RF alternatives. This ranges from very simple RF to more complex 2-way network protocols as UPnP.

Problems or disadvantages overcome by the invention

Universal remote controls need to become more powerful and more intelligent to be able to keep up to the exploding number of combinations of AV equipment and control protocols. This easily leads to complex universal remotes.

Keeping both the configuration (set-up) and usage of universal remotes simple is the main goal of this invention.

Some feature(s) of the invention

The more information the universal remote has about the CE equipment with different control protocols, the easier it becomes for a user. Especially the information on how different devices can be connected is crucial.

The core of this invention is the availability of a 'connectivity database' that is used to make the set-up & usage of the remote control simpler. Using this information in the db in a smart way actually helps people in their activity of interacting with CE equipment. The level of interaction rises from 'controlling with devices' (like TV, VCR, ...) to 'controlling activities' (watch TV, watch a movie on DVD, listen to Internet radio,).

Detailed description of how to build and use the invention

To make interaction with CE equipment easier, the connectivity database consists of multiple parts. These parts will be described in more detail below.

Support for IR CE equipment connectivity

As a first step, the universal remote needs to know the correct IR codeset of all IR devices. Multiple ways for this already exist.

The next step is knowing how IR devices are interconnected. This connectivity is needed to know how the CE equipment 'works together' to support activities like 'Play a DVD', 'Listen to Internet music', ...

I.e. to play a DVD, the TV needs to be on and switched to the right external input (EXT-2), the DVD needs to be on and in the playing mode. These IR commands have to be send to these devices, in the correct order. Also after switching the TV on, it needs some time before it will listen to new IR commands. Different TVs behave differently, different timings between IR commands, different ways to switch to external inputs, different ways to switch to a certain TV channel. The knowledge people normally have to do this right, will be loaded into the connectivity DB. Only then, the universal remote becomes 'intelligent' enough to be able to do all this automatically.

Below is a non-complete list of potential meta-data that is needed. The needed meta-data depends on the characteristics of the device.

For all devices:

- The unique identifier of the device. This usually is the model-number of the device. Alternatively the model-number of the original remote control of the device can be used to identify a device.
- Way to control power (discrete or toggle)
Discrete means a certain code always turns the device on or off. (Pressing off twice, still keeps the device off). Toggle means a certain code toggles the device on/off.
- Delay required between IR codes (after power on command and in normal operation)
Knowing the minimum time between subsequent IR-commands, can optimise the time a macro needs (quick macro). Not knowing this implies choosing safer (slower) delays between IR-commands, leading to slow responses.

For devices allowing source selection :

- Number of external input sources
- Type of the external input sources: COAX, Scart, cinch, S-Video, composite, DVI, Antenna, ...)
- Names of the inputs (AV1, AUX1, Tuner, DVD, Ext-3,....)
- Way to select the various input sources: cycling, discrete or other commands

Discrete means for each external input, there is an explicit IR command. Cycling means that a certain button will cycle through the inputs one-by-one. Also, some TVs might only go to an external input by using the program-up/program-down key (like browsing through all channels).

- In case of cycling:
 - default state at power on (=always one state or last used before power off)
 - way (command sequence) to put in a known state
 - code to use to cycle through the inputs (and required delay between codes in case different from device delay)
- In case of discrete commands:
 - Default state at power on (=always one state or last used before power off)
 - Control codes to select each of the inputs
- Alternative method.

- I.e. go to channel 0 and press program-down.

For devices allowing channel selection:

- Default channel at power on (=always one specific channel or the last used before power off)
- Method to select channels:
 - 2 digit entry
 - 3 digit entry
 - 4 digit entry
 - usage of enter key
 - usage of 10+; 100+ keys
 - -/- -
 - ..
 - Combinations of the above

For devices with teletext:

- How to turn TT on/off

Non-control related metadata (optional)

This information can be used to help selecting a certain device from the DB.

- Ranking of control codes according importance (most frequently used are ranked highest) – this may be a generic ranking per device type
- Market share per device type / manufacturer / model
- Introduction date of the product
- Picture of the product (such that user can recognize visually during set-up)
- Picture of the back panel of the product (showing connectors and printing – helps user when specifying device interconnectivity)
- Type number(s) of the remote control(s) bundled with the products
- Picture of original remote control(s) bundled with the product (such that user can recognize visually during setup)

Support for UPnP CE equipment connectivity

In case of UPnP enabled devices, the UPnP protocol supports discovering a unique device identifier per devicetype. Given that devicetype, the connectivity db will support meta-data like:

- In case it supports IR: what codeset and IR-functions
- What kind of UPnP device is it:
 - UPnP bridge (needs to be connected to a TV, Amplifier, ... to be able to play)
 - UPnP enabled TV, DVD, Amp, mini-set,
- How to turn the device on/off (UPnP and or IR)
- In case the device has inputs:
 - How many?
 - Input types.
 - Input names

- How to go to an input (UPnP and/or IR)
- What IR and/or UPnP commands exist for all features. Will remain similar to IR features.
- In case of a TV, how to switch to a channel (UPnP and/or IR)

Applications of the invention

This invention can be used for all kinds of universal remotes. Depending on the features a subset of the connectivity db will actually be needed.

Most probable application are (touch-screen) LCD remotes (like Pronto, iPronto, ...). These provide a better way to interface with the customer.

Possible extensions

1. If a meta-data information of a certain device is not in the connectivity database, the user has the means to add the required information manually. (e.g. through a wizard).
2. This manually added meta data can be added to the connectivity database and can through some means be sent to Philips.
3. The connectivity database can be implemented as an online service. The Internet connected device has the means to read from the database, and (optionally) send manual extensions to that service.
4. If manual extensions are made and stored locally on the device, these settings may reach philips through a generic 'backup configuration' services, optionally accessed through a PC.

Prior art

In some advanced remote controls like Intrigue's Harmony remote, a setup-procedure let's users setup all CE devices they have in their house. The Harmony remote will be configured to work with those devices. Other remotes may also have such a setup. When a user selects the device-type he has (TV, VCR, DVD, Home Cinema, X-Box, PC,) he is able to choose from a (prioritized) list.

RCS-5 Non Disruptive Activity Switching on Remote Control

Abstract: Intuitive mechanism that only sends RC codes to devices when a user clearly indicates he wants actively to use them

Problem

Activity based Universal remote control (with LCD and/or touchscreen)

References: UEI Nevo (www.mynevo.com), Harmony (www.harmonyremote.com)

The RC implements a user interface based on activities or tasks such as "Watch TV", "Listen Radio", "Watch DVD".

Issue is when switching from one activity to another, devices need to be switched accordingly (on, off – changing inputs etc). To do this, some IR codes need to be sent as well. Question though is *when* to send the codes.

Immediate switching may disrupt user!

Example 1 - User is watching DVD. Next he wants to consult EPG that is listed under the "Watch TV" user interface screen on his remote control, but wants to continue to watch his DVD in the mean time. RC should not switch inputs of TV in this case

Example 2: When a user is watching DVD but now wants to rewind his VCR. He uses UI of his RC to switch to activity "watch VCR". Again, the TVs input should not be changed, as the user clearly wishes to continue watching his DVD

Solution

The source switching macro should not be executed upon a user selects a new activity on the GUI of his remote control, but only later on, when a user starts actively using the new device combination.

Example 1 – Only tune TV to the internal tuner if a user has selected a show in the EPG grid and pressed a "watch" button; or only when user presses channel up.

Example 2 – Ignore VCR-rewind command, do not switch TV input. Only switch TV input when user presses "VCR-play".

Underlying technology to realize solution:

- state variables to keep track of device stats and remote control Gui state; state variables should be updated as part of macro execution under buttons
- macros or scripts under buttons rather than simple IR commands – with a conditional nature

Example macro under "VCR Play" button:

```

If TV_input_state = VCR then
  Send VCR-play command
Else:
  Send command to set TV input to VCR (eg IR code for "ext 1")
  Set TV_input_state to VCR
  Send VCR play command

```

Extension 1: power state of VCR should of course be taken into account.

```

If VCR_power_state is NOT ON
  Send VCR-On command
  Set VCR_power_state to ON

```

The problem that we have solved is to shorten this process by providing a true shortcut for selecting/starting A/V content enjoyment activities.

Solution

This invention describes a method to easily access these shortcuts to preferred activities ('presets'). We also provide a method and syntax for storing these presets. The activities that we consider are aimed at the control of A/V content retrieval and consumption, including the underlying control of legacy and networked devices

- "*Watch*": activities related to visual perception, e.g. watch a DVD on the TV;
- "*Listen*": activities related to auditory perception, e.g. listen to a PC-residing mp3 file on a legacy HiFi system (via the intermediary of a media streamer);
- "*Play*": activities related to gaming.

Elements of the invention:

1. syntax

The activities all obey the same scheme:

- they have a certain nature: Watch, Listen, Play
- they establish an association between a source and a renderer.

Hence, they can be represented by the following syntax:

<activity><source>on<renderer> 'Source' can be:

1. A physical device
2. A URI (path name of a PC directory, web URL)
3. A TV channel number or station call sign

2. storage

The system that we propose allows to store activities that have been established by the user, by retaining in memory the (conditional) macro of commands to be executed to start the source device, start the renderer, select the appropriate content item, start the actual rendering.

The instruction to store shall be generated by the user (by (long)-pressing a key, by pressing a dedicated touch screen area, or by voice command), or by the control device itself as a result of an algorithm (e.g. when a counter that counts number of times a certain <activity><source>on<renderer> combination has been used by a user passes a threshold value)

3. access

The remote control's UI will show the user an enumeration of presets. When the user selects a preset, the corresponding (conditional) macro is retrieved from memory and is executed. That is, the sequence of commands is sent, by means of the appropriate protocols and in making use of the appropriate physical transport mechanisms, to the applicable source, renderer, and possibly linking devices that are intermediary in the channel through which content will stream from source to renderer.

Examples of Presets:

- "Watch DVD on Flat Screen TV";
- "Listen to MyPC/.../MyPartyMusic on Living Room Speakers" (MyPC/.../MyPartyMusic is the directory tree on the PC);

- "Watch BBC1 on TV".

Below are a few examples of how presets effectively provide a shortcut as compared to the normal process of starting an A/V content enjoyment activity:

- "Watch ch12 on TV"

Activity flow (4 steps):

- press "Watch"
- select "TV"
- Press "1"
- Press "2"

Presets flow (1 step):

- Press "Watch ch12 on TV"

- "Watch DVD on TV"

Activity flow (4 steps):

- press "Watch"
- select "TV"
- select "DVD"
- Press "Play"

Presets flow (1 step):

- Press "Watch DVD on TV"

- "Listen to XXX.mp3 on Speakers"

Activity flow (5 steps):

- press "Listen"
- select "PC"
- select "All songs"
- (scroll to "XXX.mp3")
- Press "Play"

Presets flow (2 steps):

- (Scroll to preset "Listen to XXX.mp3 on Speakers")
- Select "Listen to XXX.mp3 on Speakers"

Possible extensions

Along the same lines of reasoning, but with a different syntax, other control tasks can be stored and accessed via the presets method. These control tasks could for example be in the area of home control and home automation:

- Control of lighting gear;
- Control of climate conditioning devices;
- Control of / access to security cameras;
-

Benefit of the invention

Main benefit is that the presets method provides a very short navigational route for users to launch A/V content enjoyment activities. This is particularly relevant when users get used to the activity-based control concept and get annoyed by having to go through the same sequence of steps time and time again.

PHNL040136EPQ

17

02.02.2004

Possible Embodiments

Any remote control application (hardware product, or computer software (PC, PDA, ...))

RCS-7 Business method for cost-efficiently creating market hype for screen-based products

Abstract: The described method provides a cost-efficient way for upgrading products, by replacing the product's skin, as enabler for marketing campaigns.

Background of the invention

Skins

Skinning is defined as the ability to change the appearance of all applications on a screen-driven product with respect to background, button bitmaps, colours, fonts and sounds. A skin is a fixed combination of the above, i.e. the user will not be allowed to do customisation by 'cherry picking'. The number of buttons, button positions, button text labels, etc... are not elements of the UI skin.

Software upgrade internet service

The notion exists of software upgrade internet service: see CRS section 10.3. As users can upgrade data partitions of the device's software, they can change the data file(s) that describe the skin used.

Solution

By making new skins available on special occasions (e.g. Halloween, Christmas period) a market hype can be created that draws a lot of attention to the product, the sole expense being defining the new skin(s) and making it available via the software upgrade internet service (that is there anyway). This may prove a very cost efficient way for enabling a marketing campaign.

Possible extensions

Extend the idea to fully customisable UIs, e.g. like in Pronto.

Benefit of the invention

Cost-efficient way to 'upgrade' a product as enabler for marketing campaigns.

Possible Embodiments

Any remote control application (hardware product, or computer software (PC, PDA, ...))

RCS-8 Method for optimised navigation in remote control applications

Abstract: Method that allows more efficient and intuitive navigation in remote control applications. It relies on the fact that navigation can be optimised, by combining notions of viewing history and hierarchical levels when moving forward and backward.

Background of the invention

Pronto-like remote control units make use of device-related control screens

Problem

The problem that arises is that the number of combined control screens or device control screens rapidly grows large (6-7 different screens), which may make navigation tedious, even when moving backward and forward with dedicated hard or soft keys. Furthermore, changes of hierarchical level, when using back and forward buttons, happen 'just' as part of the flow.

Solution

This invention describes a method to more easily and efficiently/quickly navigate remote control applications without sacrificing intuitiveness. The navigational mechanism shall cater for the concept of Smart Navigation. That is, screens will hold back/forward buttons to allow the user to quickly move back and forward through the series of screens he/she has been viewing (the "viewing history"). Moving back and forward through the viewing history happens hierarchical level per hierarchical level. This situation is depicted in below figure. Step 1 till 3 take the user from the 'home page' of the remote control application to the 2nd of six combined control screens. Pushing the *Back* button now will return the system to the list of 'watchable' devices, and not the first of six control screens. That is pushing *Back* goes to the next higher hierarchical level. Subsequently pushing the *Forward* button brings the second of six control screens on again. Repetitively pushing the *Back* button brings on the control application's home page, that is the highest hierarchical level. Further pushing the *Back* button will not resort any effect.

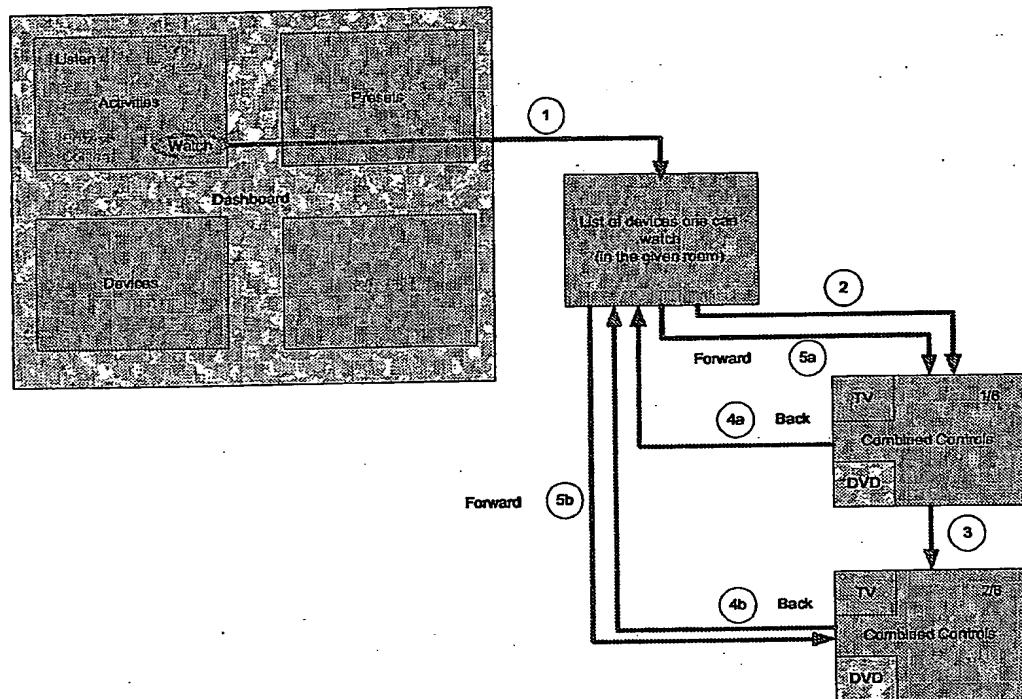


Figure: Smart Navigation Concept

Benefit of the invention

The described method combines the notion of hierarchical levels in a navigational flow with the notion of viewing history. This way the user can more quickly move backward and forward (first benefit) through the various screens, while maintaining a good feel of where he/she is in the application (second benefit).

The notion of forward and backward navigation has already been applied in Pronto remotes, however due to the lack of notion of hierarchy, only the viewing history was used making navigation much longer (e.g. going back from 'screen 6 of 6' to the next higher level takes six back-button presses, or switching to another navigational means which also lacks intuitiveness) and is far less intuitive (i.e. a user has no feeling of levels: if the sequence was DVD2→all devices→TV1→TV2, moving backward 'suddenly' changes hierarchical level after the second back-button press).

Possible Embodiments

Any remote control application (hardware product, or computer software (PC, PDA, ...))

RCS-9 Device configuration backup service

Abstract: Connected devices have many features with many settings. Also some of them are able to get software upgrades when already in the market. Since these devices are connected, for users it's very convenient to be able to back-up their configuration settings to an Internet service, PC, or other host. In case the settings get lost, wrong new settings are made, settings are lost due to a software (sw) upgrade, ..., the backup of the personal configuration(s) can be restored to the device.

In addition the backup-service is able to upgrade the settings-format to the version that works with the upgraded software version.

Background of the invention

The number of connected devices (to Internet, PC, other home-devices) are going steadily. As they have many features, there will also be many settings a user has to make to configure it optimally for him. It is always possible that a device loses it's settings (user makes wrong settings, kids play with it, it got a software upgrade that erased the settings,). In that case a user would have to do all settings again.

Devices that are able to receive software upgrades when already in the field, are more likely to lose their settings, since the settings may just be overwritten with factory settings, or the software upgrade may require another type of settings-format (incompatible with the previous ones).

Especially devices as the iPronto, store a lot of settings, not only for the device itself, but since it's a universal remote, it needs the configuration data of other IR equipment in the house. The more settings, the more frustrating it would be to lose these settings.

Some Problems or disadvantages overcome by the invention

In case the settings are lost (to any reason), a settings backup-service can back up the settings on another device. This may be an Internet service, PC, other CE device with storage, any other device.

The backup service may be able to store a list of previous versions allowing the user to go back to the settings of a few days/weeks before. It may be able to store different settings for a range of users (dad, mum, kids, ...).

In case the user wants to, or the on-device settings are corrupt, the backup-service will allow to restore the backed-up settings to the device. The user may be able to retrieve the certain backup (dated, versioned, specific user, ...).

In addition, in case the device got a software update, the backup-service is able to transform the format of a certain settings-file (used by device-software version x), to new settings-file format of device-software version y.

Some features of the invention

Bulleted, these are some of the the features of the invention:

- Be able to send a back-up of the device settings to a back-up service. This may be an Internet server, any other networked server, a PC, a CE-device, any other device that can be reached and has storage.
- The device is able to retrieve a backed-up settings-file from the service, and store it. In that case it will overwrite the existing set-up file.
- Note: there might already be multiple settings-files: per user/per location/per usage-scenario/ ...
- Add-on: The backup-service may allow to store multiple backups per user/device/location/usage-scenario. When restoring, the user is able to chose from a list of backed-up settings.
 - Add-on: the user may be able to add data like name, description, ... to his back-up of settings.
- Add-on: The back-up service may be able to migrate a certain version of a settings-file, to another version. This may be needed when the software of the device is upgraded (downgraded) to a software-version that uses another format of the settings-file.
- Add-on: The backup-service may provide default settings from other (fictitious) users. This may help an inexperienced user to retrieve a settings-file that works better for him than the factory-settings.
- Add-on: If the backup-service is owned by Philips, the settings could be used to gather information on how the users have set up their device. This can be used for marketing purposes.

Detailed description of how to build and use the invention

The system works like a client/server system. The client device is able to backup its settings on a server device that is able to store it. The server device may be on the Internet, any other network, a PC, other CE-device, any other device with storage. The client and server are able to communicate over any kind of network. A communication protocol is shared between the client and the server.

Other details are already described above.

Applications of the invention

This invention can be used for any kind of device that is connected to a network and is therefore able to communicate with other devices.

I.e. an iBoard (new contemplated remote control device) has a WiFi network connection to the home network and/or the Internet. It has many settings, not only for local iBoard settings (skin, time-zone, calibration settings, users,) but since it's a universal remote control, it holds the settings of other IR equipment in the home (TV, VCR, ...) and how this equipment is connected.

An iBoard user may spend considerable time to make these settings, losing them would mean a big annoyance.

Possible extensions

See the 'add-ons' above.

Prior art

Some devices may already be available that simply are able to send a configuration file on a PC, and restore when needed.

Harmony Remotes get their configuration from a configuration-service. Still, this is not the same as this backup service.

RCS-10 Unified Activity Based Remote Control GUI

Abstract: Intuitive unified user interface for a LCD remote control to combine legacy IR control of devices with electronic program guide information and with control of networked devices through a single consistent user interface principle, based on a task or activity oriented paradigm.

Problem

Unified user interface

Activity based Universal remote control (with LCD and/or touchscreen)

References: UEI Nevo (www.mynevo.com), Harmony (www.harmonyremote.com)

Problem: big paradigm shift for users; they do not find their individual device anymore in the UI as they were used to in classical remotes.

Integration of EPG

E.g. iPronto

Too often EPG is seen as an extra application; not integrated in the RC user interface. Result: non-user friendly user experience. Users have to switch between applications, end up in a completely different UI. Results: not easy, not fast, not consistent, easy to get lost while using the product.

Integration of network controlled devices

Rise of UPnP standard + other technologies enable legacy AV equipment to play back digital media (music, pictures, movies) stored on a PC.

No remote control exists that can handle this new class of devices, and in same time control legacy.

Issue with this new functionality is that suddenly user must navigate and choose content residing on some file system – something which he was not used to do before with legacy AV equipment (e.g. DVD, VCR are single content sources: one tape, one disc)

Equipment front-panel is not suited for this.

Remote controls sometimes have small LCDs but can not replace other legacy remotes (e.g. Creative Labs, using 2-way IR or RF proprietary solutions).

Addition of these new networked products open up new activities, the challenge is to integrate directory browsing into a unified user interface.

Solution

General GUI principle

Applying UPnP renderer –source paradigm to legacy equipment.

Navigation where user selects:

1. an activity (watch, listen)
2. if multiple present: also choose a renderer (=device with screen or speaker)
3. show a combined control screen (usually multiple screens per source-renderer combination; page scrolling mechanism + page indications are to be included)

Combined control screen shows:

1. indication of active renderer + selected source
2. Controls for renderer
3. controls for selected source
4. controls to select other sources available for this renderer

Integration of EPG

Solution is to look at a TV as a device composed of a renderer (=its screen) and multiple inputs (= ext connectors + built-in tuner).

Especially by considering the built-in tuner as a separate source makes the experience more clear to user + allows to integrate an EPG seamlessly in the UI. TV screen has multiple sources, each one of them has an associated activity.

For the built-in tuner the activity is "watch TV".

The control screens that come with this activity can be any one of the following – up to the user to choose:

- page with numeric keys to switch channels
- page with EPG grid
- page with EPG list
- page with detailed info on now playing program
-

This way EPG is fully part of the activity watch TV – and is seamlessly integrated with the UI showing all buttons and info needed for the user while performing this activity.

Integration of networked devices

Solution is again to look at the new activities these new devices provide.

Example: having a networked TV (UPnP TV or a TV with a media adapter attached to it) means the user can access content stored on any media server in the user's house. By consistently applying the UI paradigm, it is obvious to represent any detected home server on the network as a possible source for the television.

In this case it does not make sense to show the media adapter as a new source; or the TV's built-in network interface – the real source is the one where the content is residing. In this case that can be multiple servers located inside or outside the home. When a user selects such a source, the activity associated with it is "consuming digital content on TV" – effectively a combination of three possibilities: watching movies, watching pictures or listening to music. The remote control GUI (control screen) in this case should show following elements:

1. Commands to control TV volume
2. Directory browsing widget (list or folder view, tree view – depending on screen size and orientation) allowing a user to:
 - a. Browse between folders and directories
 - b. Scroll through content items in a certain directory or folder – potentially with different implementations (e.g. scrolling song titles for music; thumbnail views for pictures etc)
 - c. Highlight a content item to see more info (metadata), to preview / pre-listen to it on his remote control
 - d. Initiate playback of the highlighted content item on the selected renderer – being the TV in this example

Integration of network content services

Similar to networked servers in house, a renderer could also access internet located servers – such as internet radio stations or internet picture services (to share pictures with family or friends outside the house).

Again the paradigm can be applied consistently: all subscribed services can be presented as new sources of content for a particular rendering device with networking capabilities.

Depending on the selected service a most appropriate user interface consisting of one or more screens may be displayed; but within the consistent UI framework described earlier.

Possible extensions

Whole principle can also be turned around: start with content item – or content source to start navigation; depending on selected content a list of possible rendering devices or even locations (room) is generated by the system; after which the combined control screens are reached again to allow initiation and control of playback.

Benefit of the invention

Solves the problem described earlier: one consistent UI

Can be easily extended to any AV related device or activity a user may do in his home; independent of nature of future devices or standards for content distribution or evolution in CE market (eg trend for TVs to no longer include tuners)

Possible Embodiments

Activity based Universal remote controls (handheld, probably LCD touchscreen, networking interface) combined with UpnP control point.

Connected Planet products – OSDs on TVs etc.

Examples: Pronto, iPronto, iBoard, WiFi enabled PDA with remote control software,...

RCS 11 Distributed UPnP Controller

Abstract: A UPnP Controller (like the iBoard) will often be a mobile, battery operated device. This type of embedded devices are always resource limited (battery, processing power, memory, ...). This invention distributes the UPnP Controller application over multiple devices to overcome these weaknesses.

Background of the invention

The UPnP protocol and architecture is an network-centric way of interacting with content. It distinguishes between UPnP Server (or Content Directory Service), a device (CE device, PC, ..) that contains all kinds of content like music, movies, pictures, games, The UPnP renderer specializes in playing back the content. Finally, the UPnP Controller (control point) provides the UI of the user. It allows to user to find and locate content from the different UPnP CDSs and play them back on the available UPnP renderers. It also will show the status of the rendering (i.e. music title, artist, playing time, next song, slideshow, ...).

These logical functions can be distributed on one or multiple devices. Traditionally there will be one device that acts as the UPnP controller. Since remote controls are almost always portable, also many UPnP controllers will be portable.

Problems or disadvantages overcome by the invention

Portable embedded devices have many disadvantages. The characteristics of UPnP put heavy demands on the UPnP controller. It should stay on the network to track UPnP events, have big processing power and have enough memory to deal with a large set of content.

Since many UPnP Control Points are battery operated, this can turn into a problem. In addition to the UPnP architecture implies that a control point is the only active part that handles playback of all content (i.e. start playback of the next track when the current one ends) the UPnP Control Point is expected to stay on. Remaining always on on a wireless network is very power consuming.

Philips is in the process of patenting a feature like 'context switch' to overcome this issue. By this, the embedded sends the context of content-playback to the UPnP renderer at certain times. The render will take over the UPnP control function, so the embedded controller can go to sleep.

This idea is a generalization of the 'context switch' idea, see NL 030821 European filing no. 03101949.0, filing date June 30, 2003, title: EMBEDDING A UPnP AV MEDIASERVER OBJECT ID IN A URI. This idea relates to a UPnP-compliant MediaRenderer-Control Point combination that is enabled to exploit an organizational context of a content item as represented in a UPnP Content Directory Service. To this end, the combination is enabled to receive a URI representative of a Content Directory Service description, together with an objectID representative of the content item.

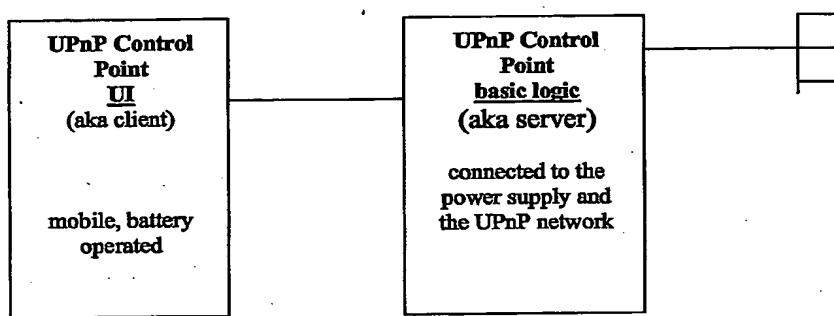
Some feature(s) of the invention

A way to overcome the limitations of the embedded UPnP controller device (showing the UI), parts of the UPnP Controller application can be off-loaded to any other device in the (home) network. For convenience, this device is called the server. In general, the UPnP controller application gets distributed. The UI runs on the

embedded device, the state in the network, playlists, actual control commands, etc. run on a server device. This server can be a separate device, or can be integrated to other devices on the network (i.e. UPnP renderer, set-top-box, PC, eHub, ...). Since the device that runs the server part can have a permanent power-connection and optionally have more resources (cpu power, memory, ...), the problems are overcome.

Detailed description of how to build and use the invention

The logical distribution of the UPnP controller functionality is distributed over 2 or more devices.



The client is connected through the server through any network (wired or wireless). (At least) the server is connected to the rest of the UPnP network. These networks may be the same or different.

The combination of the client & server make up the complete UPnP Controller application.

UPnP Control Point basic logic (server)

Keeps the main state of the UPnP network: i.e. what UPnP devices are on the network, what is currently playing, what will be the next songs, perform searches on the network, create one logical view of all content of all UPnP CDSs (content aggregation → allows to do searches over all physical servers!), etc. It connects to one or more clients to interface all this info with the user.

UPnP Control Point UI (client)

The client can either be limited to just a UI, or have its own UPnP stack. In any case, it is able to go to sleep mode (save battery) or go off. When this happens, the server will make sure the started activity (i.e. play the entire cd) is continued, and the state of the entire UPnP network is preserved.

At any time the user can wake up the client again, and resume operation.

For this to work, the client and server talk a common protocol. This protocol may be based on UPnP or anything else.

Applications of the invention

This invention can be used to make UPnP Controller that works in a distributed fashion. It allows to make a light(er)-weight client embedded device to show the UI that works with a (more powerful) server that is connected to power and the UPnP network.

CLAIMS:

A method of configuring a remote control device, the method comprising using a connectivity database, that contains a range of meta-data related to remotely controlling CE equipment; and enabling the database to drive the configuration of the remote control device.

PCT/IB2005/050380

